Computer Orientation

Bioinformatics *Baxevanis and Oullette eds. Chapter 1. pp.1-17.*

Computers and Biology

Biology has traditionally been one of the less computationally intensive sciences. The tremendous wealth of data made available by the revolution in DNA sequencing and biotechnology has fostered a marriage between biology and computer science. This is the emerging field of bioinformatics. It is worth noting that the sequence data revolution has been possible only because of the parallel revolution in information technology including advances in computer hardware and the expansion of the internet and the World Wide Web. Researchers in all branches of biology need to become more computer savvy.

The Biologist and the Internet

The desktop computer of most biologists is connected to a worldwide network of computers commonly called the **internet**. In addition to fostering communication among biologists throughout the world, the internet allows access to in increasing number of important biological databases. Internet access is essential because it allows researchers almost instantaneous access to data from distant sites and allows computational access to datasets that are too large and expensive to maintain and manipulate as local copies.

Access to internet servers can be through electronic mail or standard internet protocols such as file transfer protocol (**ftp**). Requests are routed to the correct server through its unique address called an Internet Protocol address or **IP address**. All computers on the internet have a unique IP address. The format of an IP address is a 32-bit numeric address written as four numbers separated by periods. Each number can be zero to 255. For example, 130.14.22.106 is the IP address of one of the front-end web servers at the NCBI.

The World Wide Web

Internet servers that can handle documents formatted in hypertext markup language (**HTML**) are part of the World Wide Web (**WWW**). **HTML** allows links to other documents, as well as graphics, audio, and video. Web browsers are applications that make it easy to access the World Wide Web. The two of the most popular web browsers are Netscape Navigator and Microsoft's Internet Explorer. Web browsers can access these HTML documents through hypertext transfer protocol (**http**). The complete address for accessing a resource on the WWW is called a Uniform Resource Locator (**URL**). The URL consists of a protocol followed by IP address or **domain name** of the computer holding the documnet and finally the file name of the specific document. For example

`http://www.ncbi.nlm.nih.gov/index.html`

retrieves the NCBI main (or home) page. In this example the NCBI web server is identified by its fully qualified domain name instead of its IP address. The internet really relies on IP addresses, but humans are better at remembering domain names. A network called the Domain Name Service (DNS) maintains the correspondence between domain names and IP addresses.

Computer Basics

Everyone is at least loosely familiar with the concepts of hardware and software: hardware consists of the wires and other electronic gizmos that are the guts of the computer; software is an instruction sets that on a good day makes the computer do something useful or maybe entertaining. The most important piece of software on a computer is the operating system (OS). The OS manages and holds together all of the other components of the computer, hardware and software. It oversees such tasks as accepting input and writing output, allocating memory and accessing fixed disk storage managing files and directories. We often use the term **platform** to describe the operating system. This in turn is built for on the instruction set for a processor or microprocessor, the hardware that performs logic operations and manages data movement in the computer. The operating system must be designed to work with the particular processor's set of instructions. For example, Microsoft's Windows 2000 is built to work with a series of microprocessors from the Intel Corporation that share the same or similar sets of instructions.

The UNIX operating system

Most people are familiar with the common desktop computing platforms: Intel based PC running Windows and Macintosh computers running some version of the Mac OS. Another operating system that is much more important in bioinformatics efforts is UNIX. UNIX was developed at Bell Labs in the 1970s as a flexible multi user, multitasking operating system for programmers. It power and flexibility make still the most important OS for development, and it is the OS of choice for any serious bioinformatics tool or database development, and runs the servers at the important biological database web or internet sites. The UNIX operating system was originally written in the C language and is still available as open source. Many computer manufacturers have also ported the UNIX operating systems to their system architectures. Notable examples are Solaris for Sun Microsystems, OSF1 for Compaq (formerly DEC), IRIX for Silicon Graphics. LINUX is a UNIX OS for Intel based machines (PCs) based on an open source model of software development. LINUX is becoming very popular because of the inexpensive cost of the OS (nearly free). More important is the increasing power and the lower cost of Intel based hardware, which now rival the capabilities of the systems mentioned above — what computer jocks call “real computers”— but for less money.

Access to UNIX systems

The main mode of access to a UNIX machine is through a *terminal*, which usually includes a keyboard, and a video monitor. For each terminal connected to the UNIX system, the main UNIX program (the kernel) runs a process called a *tty* that accepts input from the terminal, and sends output to the terminal. Every UNIX system has a main console that is connected directly to the machine. Some terminals are referred to as "dumb" terminals because they can only send characters as input to the UNIX system, and receive characters as output from the UNIX system. PCs are often used to emulate dumb terminals, so that they can be connected to a UNIX system. Xterminals support graphical user interfaces with mouse interaction, icons, windowing systems and menus similar to those found on Macintosh and Windows desktop machines. The Common Desktop Environment (CDE) on Solaris provides such a graphical user interface.

The UNIX shell

The shell is a special program that is the interface between the user and the main UNIX program (the kernel). The shell functions as the command interpreter. There are several different kinds of shell with somewhat different command languages. Common shells are the Bourne (again) shell, the Korn shell and the C shell (csh). We will be working with the C shell in this course. At the terminal the shell presents the user with a command prompt, which indicates that the shell is ready to accept a new command:

```
ray>
```

There are some special operators recognized by the shell that are useful.
The greater than sign (`>`) serves as the redirection output operator. It is often used to redirect output to file instead of the standard output.

```
ray> ls > directory.listing
```

will direct the directory content listing to a file called `directory.listing.`

The “pipe” (`|`) is used to send the output of one shell command or program as input to another.

```
ray> ls | grep txt
```

will report all files and directories with the string txt in their names

The ampersand (&) specifies that a process run in the background. Use this when launching graphical user interface program from the shell command line.

```
ray> netscape &
```

The shell is case sensitive. A file called Myfile is different than a file called myfile. Likewise Grep is not the name of a utility; grep is.

The UNIX File System

All the stored information on a UNIX computer is kept in a filesystem. The UNIX filesystem is hierarchical (resembling a tree structure). The tree is anchored at a place called the root, designated by a front slash "/". Every item in the filesystem tree is either a file, or a directory. A directory can contain files, and other directories. Whenever you're interacting with the system, the shell considers you to be located somewhere within the filesystem. The place in the filesystem tree where you are located is called the *current* working directory. Typing `pwd` at the command prompt will show the path from the root to the current working directory. In the path produced the nodes or levels in the tree are separated by the front slash. For example

```
ray> pwd

/ray/web/public/htdocs
```

A double period is used to specify the parent directory.

```
ray> pwd

/ray/web/public/htdocs

ray> cd ..

ray> pwd

/ray/web/public
```

To return to your home directory, type `cd` without a directory name and hit return.

UNIX utility programs

There a large number of built-in powerful and flexible programs that perform various tasks including monitoring the system, navigation and creation of files and directories. All of these programs can be run from a terminal window by typing the name of program and pressing the enter key. Many of these utilities take various kinds of command line options (also called arguments). The utility **`man`** provides a detailed description and the various command line options (arguments) For example

```
>man cp
```

The problem is that you have to know the name of the command to use this. **Table 1** provides a list and a brief description of frequently used utilities and shell commands.

Table 1. Common UNIX Utilities and Shell Commands.

cat	Dumps the contents of a file to standard output
cd	Change directory
chmod	Change file permissions to make a file readable, write able or executable (or not) to yourself, your group or the whole world
compress [filename]	The UNIX file compression utility. Reduces file sizes and replaces the original file with file.Z. Achieve about 2-3 fold reduction in the sizes of text files.
cp [filename] [filename]	Copy. Copies a file to another location
grep	Get Regular Expression: finding strings in files
gunzip [filename.gz]	Undoes gzip compression (*.gz). Gzip also will undo standard UNIX compression (*.Z), but uncompress will not work on gzipped files.
gzip [filename]	Another UNIX compression utility. Adds the extension .gz to the compressed file.
head [filename]	Dumps the top few lines of a file to the terminal screen
kill [process id]	Kills a process. Use kill –9 to show no mercy
less [filename]	Like more only less.
ln [link] [filename]	Create a link between a name and a file or directory
logout	Disconnect from the machine
ls	Directory listing
mkdir [dirname]	Create a directory
more [filename]	Dumps the contents of a file to the terminal screen -- one screen at a time. Hit the space bar to display the next screen.
mv [filename] [filename]	Move. This is really a way to rename a file.
phone	Gives names usernames and phone number for people on the network
ps	List processes that are running
pwd	Provides current path.
rlogin [hostname]	Establish a login shell to another machine on the network
rm [filename]	Remove. Deletes a file
rmdir [dirname]	Removes (deletes) a directory
source [filename]	Causes the shell to read the contents of a file.
tail [filename]	Dumps the last few lines of a file to the terminal screen
tar	Tape Archive: Creates, updates or extracts files from a tape archive UNIX tar files are commonly seen on ftp where they bundle sites multi-file packages to simplify downloads. simplify downloads. Common usage: tar –xvf Extracts all files from a tape archive.
top	Lists the most CPU intensive processes running on a machine.
uncompress [filename.Z]	Undoes UNIX compression.
xkill	The XWindows equivalent of kill. Allows killing a process by using the mouse to click on the window it created

Special Protocols

The following special protocols are available from the shell.

`ftp` Allows the internet file transfer protocol. The domain name or the IP address of the ftp server can be passed on the command line. Once at the ftp prompt type help to see a list of commands.

`telnet` An internet protocol to establish a dumb terminal connection to a UNIX system. This protocol is most useful as a way to connect from a PC to a UNIX system. The Windows OS comes with a telnet program. This protocol is no longer allowed to connect to the NCBI machines from outside the network because of security issues.

Solaris software

The Solaris operating system comes with a number useful XWindows style programs that run under the CDE. These can be launched from a menu on the desktop or from a terminal window by typing the name of the program

Third party software

There are a few third party programs available on the NCBI networks that are useful enough to mention here. **pine** is the University of Washington mail program which is based on the screen text editor **pico**. Pine is very useful for managing email when accessing the system through a dumb terminal. Typing `pico` or `pine` at the command prompt will launch these programs. The program nedit is a full-featured Xwindows style text editor. Launch this with an ampersand from the shell (`nedit &`).

Exercises:

1. The shell manages which files on the network are accessible by reading from a series of scripts at login. These scripts are "hidden" files. These are also called "dot" files because their names begin with a period. To see the hidden files use the ls utility with the -a option. List all files in your home directory. View the contents of the contents of the `.login` file using `cat`. The `.login` file refers to the `.ncbi_hints` file. Open this latter file using pico or nedit. If necessary, edit the facilities line to include `biologist` and `ncbi_tools`. You will need to source the `.login` file to make these changes take effect.

2. Use the `host` utility at the shell prompt to find the IP address for **www.ncbi.nlm.nih.gov**. Verify that using this IP address or the domain name in a URL with Netscape retrieves the same web page.

3. Create a directory called Sequin in your home directory. Use command line ftp to connect to the NCBI anonymous ftp server. Change to the sequin directory then to the CURRENT directory. Download the sequin package for Solaris. Put it in the Sequin directory that you created. Uncompress and extract the archives then run the program from the shell.